

PART B UNIT 6

NATURAL LANGUAGE PROCESSING



CLASS 10
ARTIFICIAL INTELLIGENCE (417)
INDIAN SCHOOL AL WADI AL KABIR



Natural Language Processing

Computers require a specific set of instructions to understand human input called programs. To talk to a computer, we convert natural language into a language that a computer understands. We need Natural Language Processing to help computers understand natural language.

Why is NLP important?

Computers can only process electronic signals in the form of binary language. Natural Language Processing facilitates this conversion to digital form from the natural form. Thus, the whole purpose of NLP is to make communication between computer systems and humans possible. This includes creating different tools and techniques that facilitate better communication of intent and context.

Demystify Natural Language Processing (NLP)

Natural Language Processing, or NLP, is the sub-field of AI that is focused on enabling computers to analyse, understand and process human languages to derive meaningful information from human language.

Wordtune (AI writing tool that rewrites, rephrases, and rewords your writing)



Natural Language Processing

Applications of Natural Language Processing:

Automatic Summarization:

Information overload is a real problem when we need to access a specific, important piece of information from a huge knowledge base. Automatic summarization is relevant not only for summarizing the meaning of documents and information, but also to understand the emotional meanings within the information, such as in collecting data from social media.

Sentiment Analysis:

The goal of sentiment analysis is to identify sentiment among several posts or even in the same post where emotion is not always explicitly expressed.

Companies use Natural Language Processing applications, such as sentiment analysis, to identify opinions and sentiment online to help them understand what customers think about their products and services

Text classification:

Text classification makes it possible to assign predefined categories to a document and organize it to help you find the information you need or simplify some activities. For example, an application of text categorization is spam filtering in email.

Autogenerated captions:

Captions are generated by turning natural speech into text in real-time. It is a valuable feature for enhancing the accessibility of video content. For example: Auto-generated captions on YouTube and Google Meet.

Language Translation: It incorporates the generation of translation from another language. This involves the conversion of text or speech from one language to another, facilitating cross-linguistic communication and fostering global connectivity. For example: Google Translate

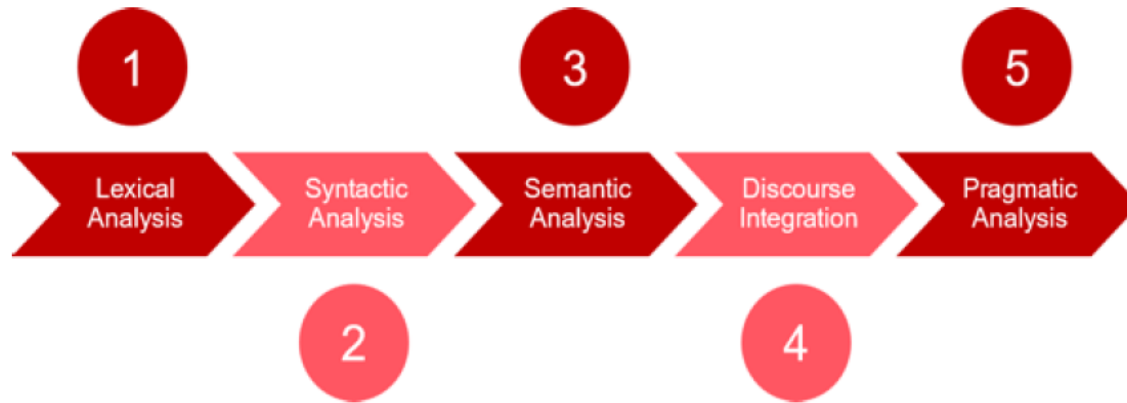
Keyword Extraction: Keyword extraction is a tool that automatically extracts the most used, important words and expressions from a text. It can give valuable insights into people's opinions about any business on social media. Customer Service can be improved by using a Keyword extraction tool.

Virtual Assistants:

Accessing our data, helps us in keeping notes of our tasks, make calls for us, send messages and a lot more. With the help of speech recognition, these assistants can not only detect our speech but can also make sense out of it. Ex: Google Assistant, Cortana, Siri, Alexa, etc

Stages of Natural Language Processing (NLP)

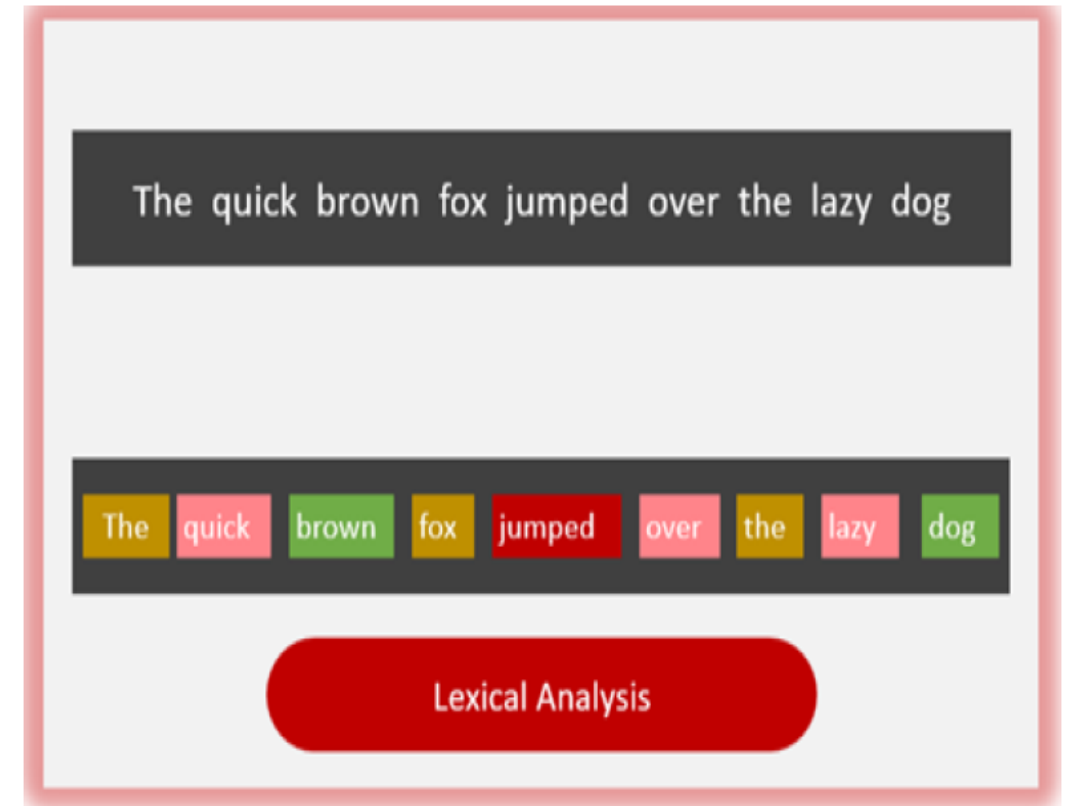
The different stages of Natural Language Processing (NLP) serve various purposes in the overall task of understanding and processing human language. The stages of Natural Language Processing (NLP) typically involve the following:



1. Lexical Analysis:

NLP starts with identifying the structure of input words. It is the process of dividing a large chunk of words into structural paragraphs, sentences, and words. **Lexicon** stands for a collection of the various words and phrases used in a language.

Lengthy text is broken down into chunks.



2. Syntactic Analysis / Parsing

It is the process of checking the grammar of sentences and phrases. It forms a relationship among words and eliminates logically incorrect sentences.

The grammar is correct!

The quick brown **jumped fox** over the lazy dog

Wrong

The quick brown fox jumped over the lazy dog

Right

3. Semantic Analysis

In this stage, the input text is now checked for meaning, and every word and phrase is checked for meaningfulness.

For example: It will reject a sentence that contains 'hot ice cream' in it. The fox jumped into the dog.

Sentences make actual sense!

The quick brown **jumped fox** over the lazy dog

Wrong

The quick brown fox jumped over the lazy dog

Right

Discourse Integration

It is the process of forming the story of the sentence. Every sentence should have a relationship with its preceding and succeeding sentences. The flow of words makes sense!

The quick brown fox jumped foxed over the lazy dog.
Then **it** went into the thick bushes.

Here 'it' means 'the fox'

The quick brown fox jumped over **it**. Then it went into the thick bushes.

'it' is unknown here

Pragmatic Analysis

In this stage, sentences are checked for their relevance in the real world. Pragmatic means practical or logical, i.e., this step requires knowledge of the intent in a sentence. It also means to discard the actual word meaning taken after semantic analysis and take the intended meaning.

The intended meaning has been achieved!

Relax! I'm just pulling your leg







Means he is just joking

Does not mean he's pulling his actual leg



Chatbots

One of the most common applications of Natural Language Processing is a chatbot.

	<ul style="list-style-type: none">• Mitsuku Bot* https://www.pandorabots.com/mitsuku/
	<ul style="list-style-type: none">• CleverBot* https://www.cleverbot.com/
	<ul style="list-style-type: none">• Jabberwacky* http://www.jabberwacky.com/
	<ul style="list-style-type: none">• Haptik* https://haptik.ai/contact-us
	<ul style="list-style-type: none">• Rose* http://ec2-54-215-197-164.us-west-1.compute.amazonaws.com/speech.php
	<ul style="list-style-type: none">• Ochatbot* https://www.ometrics.com/blog/list-of-fun-chatbots/

CHAT BOTS

Script- bot and Smart-bot.

- **Chatbots** also known as conversational agents, are software applications that mimic written or spoken human speech for the purposes of simulating a conversation or interaction with a real person
- **Examples of script bot** may include the bots which are deployed in the customer care section of various companies. Their job is to answer some basic queries that they are coded for and connect them to human executives once they are unable to handle the conversation.
- **Examples of smart bot :** On the other hand, all the assistants like Google Assistant, Alexa, Cortana, Siri, etc. can be taken as smart bots as not only can they handle the conversations but can also manage to do other tasks which makes them smarter.



There are 2 types of chatbots around us: Script- bot and Smart-bot.

Script-bot	Smart-bot
Script bots are easy to make	Smart-bots are flexible and powerful
Script bots work around a script which is programmed in them	Smart bots work on bigger databases and other resources directly
Mostly they are free and are easy to integrate to a messaging platform	Smart bots learn with more data
No or little language processing skills	Coding is required to take this up on board
Limited functionality	Wide functionality

AI Data Processing

- Humans interact with each other very easily. For us, the natural languages that we use are so convenient that we speak them easily and understand them well too. But for computers, our languages are very complex. **Natural Language Processing makes it possible for the machines to understand and speak in the Natural Languages just like humans.**
- **The language of computers is Numerical. So the very first step is to convert our language to numbers. This conversion takes a few steps to happen. The first step to it is Text Normalisation.**
- Since human languages are complex, we need to first of all simplify them in order to make sure that the understanding becomes possible.

AI Data Processing

Text Normalisation : In Text Normalisation, we undergo several steps to normalise the text to a lower level. Before we begin, we need to understand that in this section, we will be working on a collection of written text. That is, we will be working on text from multiple documents and the term used for the whole textual data from all the documents altogether is known as corpus. Not only would we go through all the steps of Text Normalisation, we would also work them out on a corpus. Text Normalisation helps in cleaning up the textual data in such a way that it comes down to a level where its complexity is lower than the actual data.

In Text Normalisation, we undergo several steps to normalise the text to a lower level.

The Steps are:

1. Sentence Segmentation
2. Tokenisation
3. Removing Stopwords, Special Characters and Numbers
4. Converting text to a common case
5. Stemming
6. Lemmatization



1. Sentence Segmentation

Under sentence segmentation, the whole corpus is divided into sentences. Each sentence is taken as a different data so now the whole corpus gets reduced to sentences.

In CBT, we learn to decipher the lies we are undermining ourselves with— based on the bias embedded in the things we say. For example, “I’m never going to make any friends” is an example of all-or-nothing thinking and we feel bad because we buy into this thought.



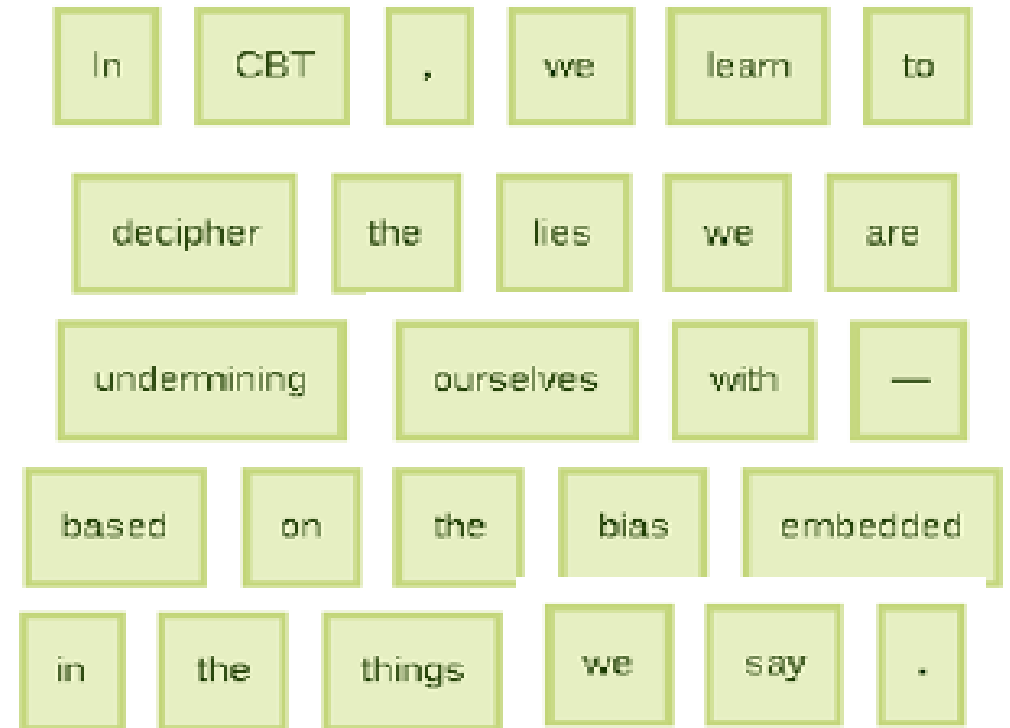
1. In CBT, we learn to decipher the lies we are undermining ourselves with— based on the bias embedded in the things we say.
2. For example, “I’m never going to make any friends” is an example of all-or-nothing thinking and we feel bad because we buy into this thought.



2. Tokenisation

After segmenting the sentences, each sentence is then further divided into tokens. Tokens is a term used for any word or number or special character occurring in a sentence. Under tokenisation, every word, number and special character is considered separately and each of them is now a separate token.

In CBT, we learn to decipher the lies we are undermining ourselves with— based on the bias embedded in the things we say.

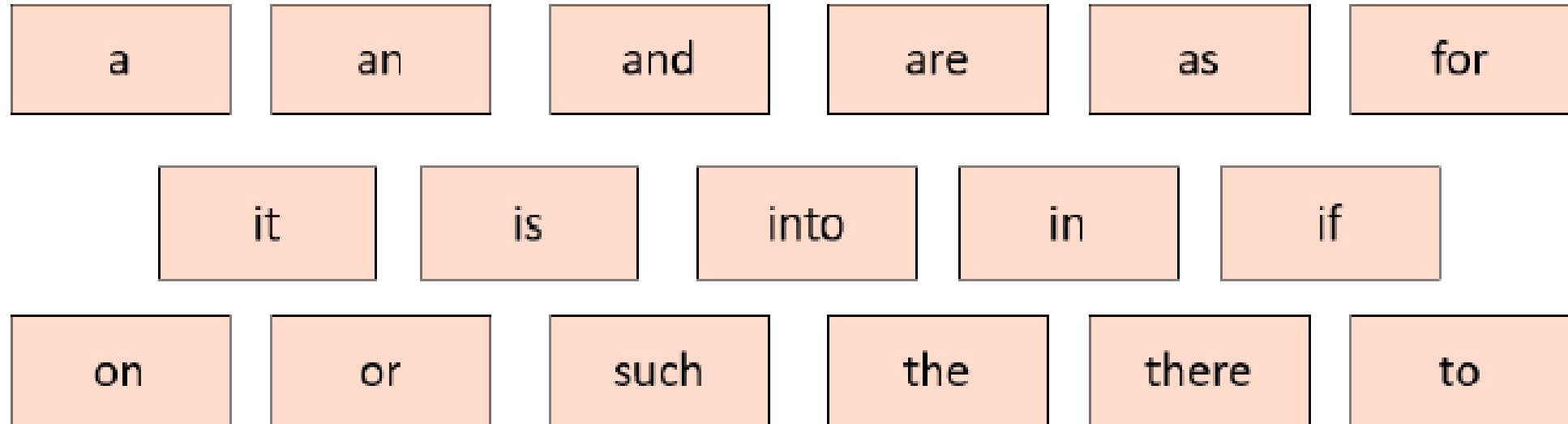




3. Removing Stopwords, Special Characters and Numbers

In this step, the tokens which are not necessary are removed from the token list.

Stopwords are the words which occur very frequently in the corpus but do not add any value to it. Humans use grammar to make their sentences meaningful for the other person to understand. But grammatical words do not add any essence to the information which is to be transmitted through the statement hence they come under stopwords. Some examples of stopwords are:

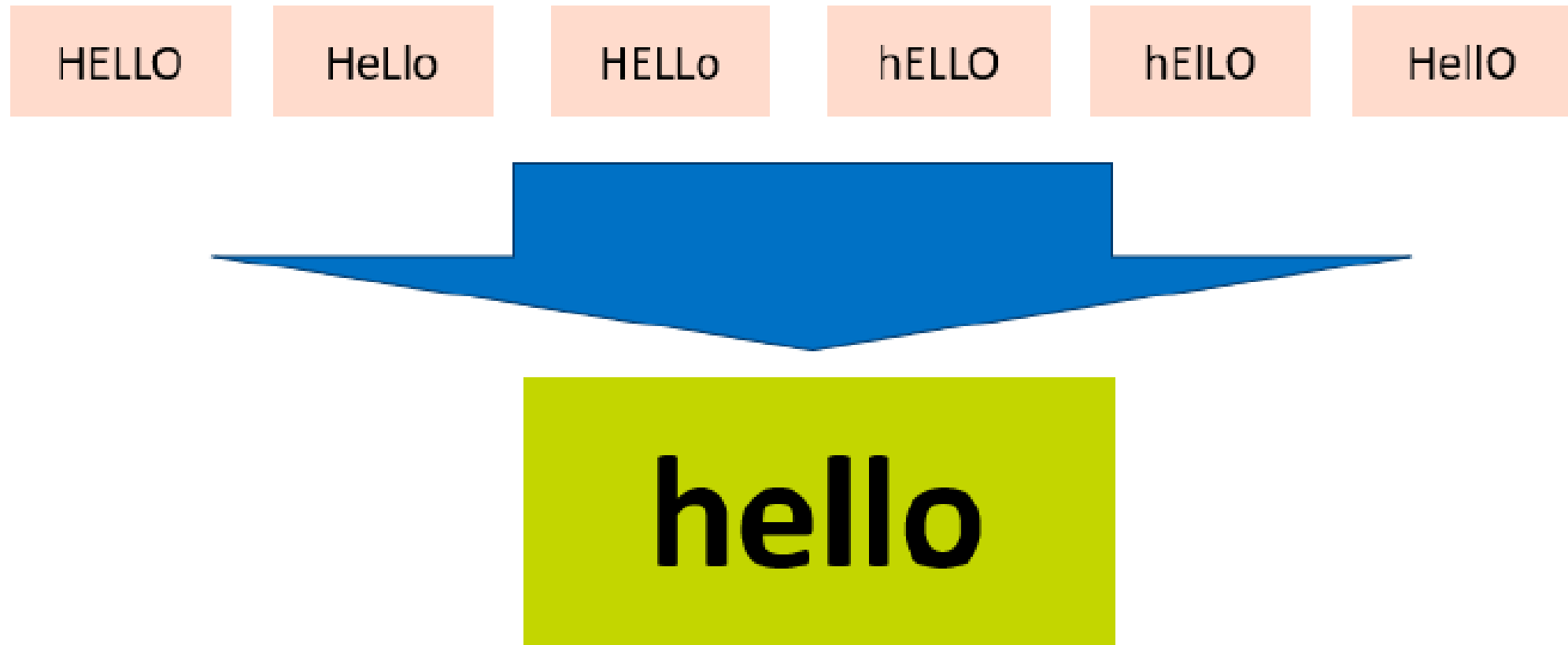


These words occur the most in any given corpus but talk very little or nothing about the context or the meaning of it. Hence, to make it easier for the computer to focus on meaningful terms, these words are removed.

Along with these words, a lot of times our corpus might have special characters and/or numbers. Now it depends on the type of corpus that we are working on whether we should keep them in it or not. For example, if you are working on a document containing email IDs, then you might not want to remove the special characters and numbers whereas in some other textual data if these characters do not make sense, then you can remove them along with the stopwords.

4. Converting text to a common case

After the stopwords removal, we convert the whole text into a similar case, preferably **lower case**. This ensures that the case-sensitivity of the machine does not consider same words as different just because of different cases.



5. Stemming

In this step, the remaining words are reduced to their root words. In other words, stemming is the process in which the affixes of words are removed and the words are converted to their base form.

Note that in stemming, the stemmed words (words which are we get after removing the affixes) might not be meaningful. Here in this example as you can see: healed, healing and healer all were reduced to heal but studies was reduced to studi after the affix removal which is not a meaningful word.

Word	Affixes	Stem
healed	-ed	heal
healing	-ing	heal
healer	-er	heal
studies	-es	studi
studying	-ing	study

Stemming does not take into account if the stemmed word is meaningful or not. It just removes the affixes hence it is faster.

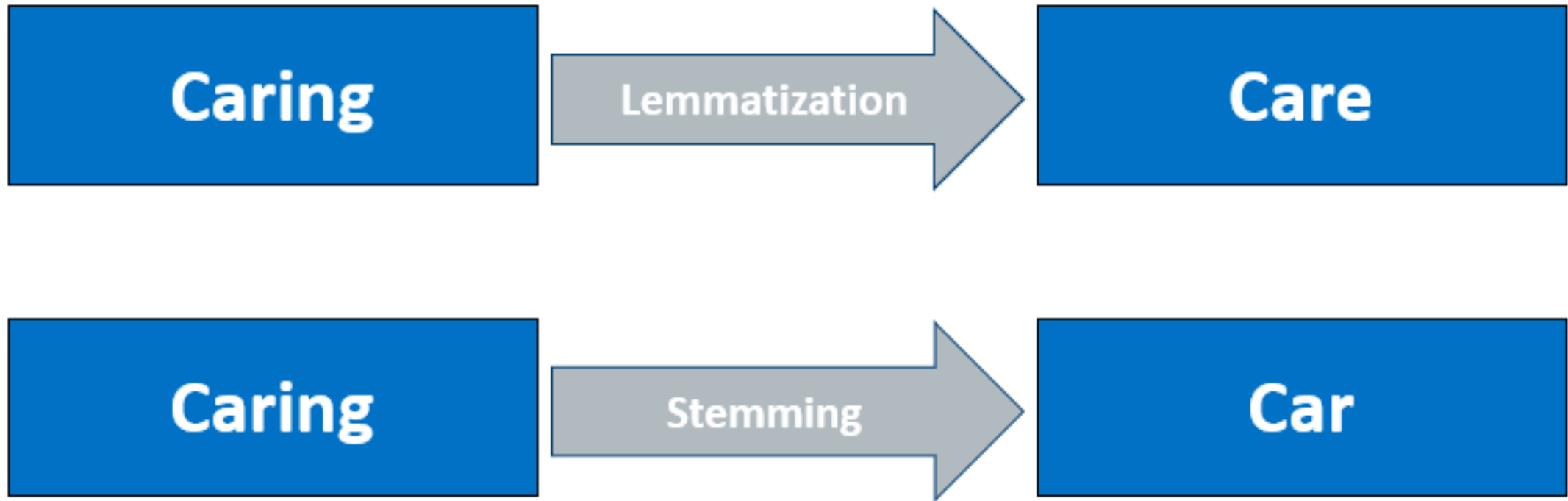
6. Lemmatization

Stemming and lemmatization both are alternative processes to each other as the role of both the processes is same – removal of affixes. But the difference between both of them is that in lemmatization, the word we get after affix removal (also known as lemma) is a meaningful one. **Lemmatization makes sure that lemma is a word with meaning and hence it takes a longer time to execute than stemming.**

As you can see in the same example, the output for studies after affix removal has become study instead of studi.

Word	Affixes	lemma
healed	-ed	heal
healing	-ing	heal
healer	-er	heal
studies	-es	study
studying	-ing	study

Difference between stemming and lemmatization can be summarized by this example:



With this we have normalised our text to tokens which are the simplest form of words present in the corpus.

Now it is time to convert the tokens into numbers. For this, we would use the Bag of Words algorithm

Bag of Words

Bag of Words is a Natural Language Processing model which helps in extracting features out of the text which can be helpful in machine learning algorithms. In bag of words, we get the occurrences of each word and construct the vocabulary for the corpus.

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

This image gives us a brief overview about how bag of words works. Let us assume that the text on the left in this image is the normalised corpus which we have got after going through all the steps of text processing. Now, as we put this text into the bag of words algorithm, the algorithm returns to us the unique words out of the corpus and their occurrences in it. As you can see at the right, it shows us a list of words appearing in the corpus and the numbers corresponding to it shows how many times the word has occurred in the text body. Thus, we can say that **the bag of words gives us two things:**

1. A vocabulary of words for the corpus

2. The frequency of these words (number of times it has occurred in the whole corpus).

Here calling this algorithm “bag” of words symbolises that the sequence of sentences or tokens does not matter in this case as all we need are the unique words and their frequency in it.

Here is the step-by-step approach to implement bag of words algorithm:

1. Text Normalisation: Collect data and pre-process it
2. Create Dictionary: Make a list of all the unique words occurring in the corpus. (Vocabulary)
3. Create document vectors: For each document in the corpus, find out how many times the word from the unique list of words has occurred.
4. Create document vectors for all the documents.

Let us go through all the steps with an example:

Step 1: Collecting data and pre-processing it.

Document 1: *Aman and Avni are stressed*

Document 2: *Aman went to a therapist*

Document 3: *Avni went to download a health chatbot*

Here are three documents having one sentence each. After text normalisation, the text becomes:

Document 1: [aman, and, avni, are, stressed]

Document 2: [aman, went, to, a, therapist]

Document 3: [avni, went, to, download, a, health, chatbot]

Note that no tokens have been removed in the stopwords removal step. It is because we have very little data and since the frequency of all the words is almost the same, no word can be said to have lesser value than the other.

Step 2: Create Dictionary

Go through all the steps and create a dictionary i.e., list down all the words which occur in all three documents:

aman	and	avni	are	stressed	went
download	health	chatbot	therapist	a	to

Note that even though some words are repeated in different documents, they are all written just once as while creating the dictionary, we create the list of unique words.

Step 3: Create document vector

In this step, the vocabulary is written in the top row. Now, for each word in the document, if it matches with the vocabulary, put a 1 under it. If the same word appears again, increment the previous value by 1. And if the word does not occur in that document, put a 0 under it.

Since in the first document, we have words: aman, and, anil, are, stressed. So, all these words get a value of 1 and rest of the words get a 0 value.

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
`1	1	1	1	1	0	0	0	0	0	0	0

Step 4: Repeat for all documents

Same exercise has to be done for all the documents. Hence, the table becomes:

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
`1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1

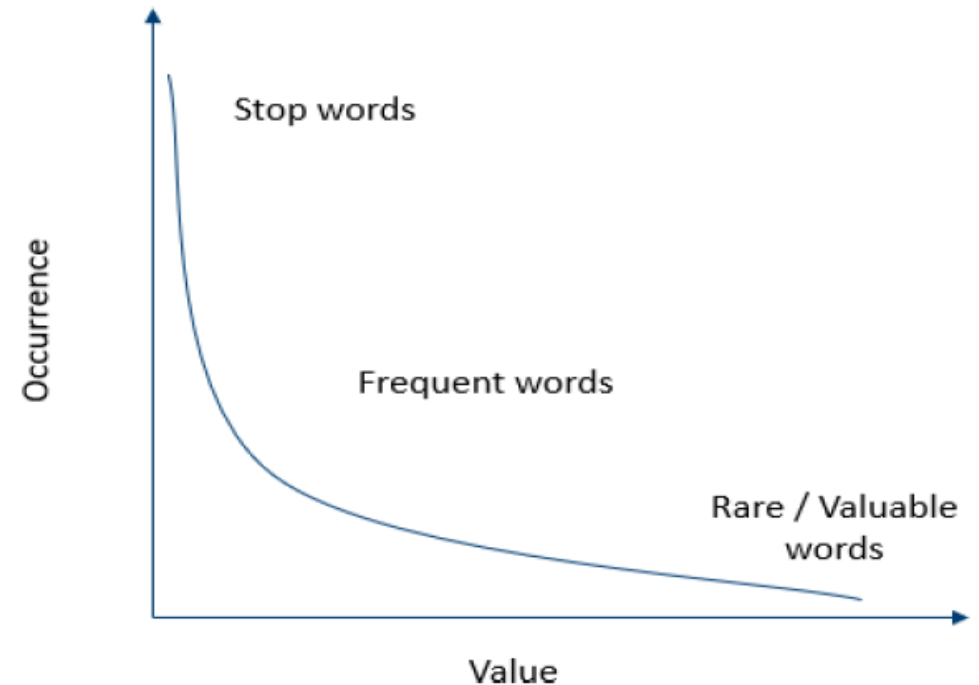
In this table, the header row contains the vocabulary of the corpus and three rows correspond to three different documents. Take a look at this table and analyse the positioning of 0s and 1s in it.

Finally, this gives us the **document vector table** for our corpus. But the tokens have still not converted to numbers. This leads us to the final steps of our algorithm: TFIDF(Term Frequency & Inverse Document Frequency)

TFIDF: Term Frequency & Inverse Document Frequency –

TFIDF helps us identify the value of each word.

- Take a look at this graph. It is a plot of the occurrence of words versus their value. As you can see, if the words have the highest occurrence in all the documents of the corpus, they are said to have negligible value hence they are termed as **stop words**. These words are mostly removed at the pre-processing stage only.
- Now as we move ahead from the stop words, the occurrence level drops drastically and the words which have adequate occurrence in the corpus are said to have some amount of value and are termed as frequent words. These words mostly talk about the document's subject and their occurrence is adequate in the corpus. Then as the occurrence of words drops further, the value of such words rises. These words are termed as **rare or valuable words**. These words occur the least but add the most value to the corpus.



Term Frequency

Term frequency is the frequency of a word in one document. Term frequency can easily be found in the document vector table as in that table we mention the frequency of each word of the vocabulary in each document.

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1

Inverse Document Frequency

Document Frequency is the number of documents in which the word occurs irrespective of how many times it has occurred in those documents. The document frequency for the exemplar vocabulary would be:

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
2	1	2	1	1	2	2	2	1	1	1	1

Here, you can see that the document frequency of 'aman', 'avni', 'went', 'to' and 'a' is 2 as they have occurred in two documents. The rest of them occurred in just one document hence the document frequency for them is one.

Talking about inverse document frequency, we need to put the document frequency in the denominator while the total number of documents is the numerator. Here, the total number of documents is 3, hence inverse document frequency becomes:

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
$\frac{1}{3/2}$	$\frac{3}{1}$	$\frac{3}{2}$	$\frac{3}{1}$	$\frac{3}{1}$	$\frac{3}{2}$	$\frac{3}{2}$	$\frac{3}{2}$	$\frac{3}{1}$	$\frac{3}{1}$	$\frac{3}{1}$	$\frac{3}{1}$

Finally, the formula of TFIDF for any word W becomes:

$$\text{TFIDF}(W) = \text{TF}(W) * \log(\text{IDF}(W))$$

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
$1 * \log(3/2)$	$1 * \log(3)$	$1 * \log(3/2)$	$1 * \log(3)$	$1 * \log(3)$	$0 * \log(3/2)$	$0 * \log(3/2)$	$0 * \log(3/2)$	$0 * \log(3)$	$0 * \log(3)$	$0 * \log(3)$	$0 * \log(3)$
$1 * \log(3/2)$	$0 * \log(3)$	$0 * \log(3/2)$	$0 * \log(3)$	$0 * \log(3)$	$1 * \log(3/2)$	$1 * \log(3/2)$	$1 * \log(3/2)$	$1 * \log(3)$	$0 * \log(3)$	$0 * \log(3)$	$0 * \log(3)$
$0 * \log(3/2)$	$0 * \log(3)$	$1 * \log(3/2)$	$0 * \log(3)$	$0 * \log(3)$	$1 * \log(3/2)$	$1 * \log(3/2)$	$1 * \log(3/2)$	$0 * \log(3)$	$1 * \log(3)$	$1 * \log(3)$	$1 * \log(3)$

Here, you can see that the IDF values for Aman in each row are the same and a similar pattern is followed for all the words of the vocabulary. After calculating all the values, we get:

aman	and	avni	are	stressed	went	to	a	therapist	download	health	chatbot
0.176	0.477	0.176	0.477	0.477	0	0	0	0	0	0	0
0.176	0	0	0	0	0.176	0.176	0.176	0.477	0	0	0
0	0	0.176	0	0	0.176	0.176	0.176	0	0.477	0.477	0.477

Finally, the words have been converted to numbers. These numbers are the values of each for each document. Here, you can see that since we have less amount of data, words like 'are' and 'and' also have a high value. But as the IDF value increases, the value of that word decreases.

Summarizing the concept, we can say that:

1. Words that occur in all the documents with high term frequencies have the lowest values and are considered to be the stop words.
2. For a word to have a high TFIDF value, the word needs to have a high term frequency but less document frequency which shows that the word is important for one document but is not a common word for all documents.
3. These values help the computer understand which words are to be considered while processing the natural language. The higher the value, the more important the word is for a given corpus.

Applications of TFIDF

TFIDF is commonly used in the Natural Language Processing domain. Some of its applications are:

Document Classification	Topic Modelling	Information Retrieval System	Stop word filtering
Helps in classifying the type and genre of a document.	It helps in predicting the topic for a corpus.	To extract the important information out of a corpus.	Helps in removing unnecessary words from a text body.